Instituto Tecnológico de Costa Rica

Área Académica de Ingeniería en Computadores
(Computer Engineering Academic Area)

Programa de Licenciatura en Ingeniería en Computadores
(Licentiate Degree Program in Computer Engineering)

Curso: CE-4302 Arquitectura de Computadores II
(Course: CE-4302 Computer Architecture II)

Especificación Proyecto II
(Project II specification)

Profesor:
(Professor)

Ing.Jeferson González Gómez, M.Sc

Fecha de entrega: 10 de Mayo de 2019
(Due Date: May 10th, 2019)

Proyecto II. Modelo de procesador vectorial para encriptación de imágenes

Índice

1. Objetivos

Mediante el desarrollo de este proyecto, el estudiante aplicará los conceptos de paralelismo a nivel de datos, específicamente procesadores SIMD del tipo vectorial, por medio del diseño e implementación en software de una arquitectura vectorial propia para el tratamiento de imágenes, en algoritmos de encriptación, como una posible aplicación de seguridad informática. Adicionalmente, este proyecto pretende ejercitar en los estudiantes los atributos de diseño en ingeniería y habilidades de comunicación, tal y como se describe más adelante.

2. Atributos relacionados

A continuación se describen los atributos del graduado que se presenten abordar con el desarrollo del proyecto.

2.1. Diseño (DI)

Capacidad para diseñar soluciones de problemas complejos de ingeniería, con final abierto y diseñar sistemas, componentes o procesos que cumplan con necesidades específicas, considerando la salud pública, seguridad, estándares pertinentes, así como los aspectos culturales, sociales, económicos y ambientales.
El atributo de diseño será evaluado tanto formativamente (reuniones de seguimiento con el profesor) como sumativamente, en especial en la sección de documentación de diseño de los entregables.

2.2. Habilidades de comunicación (HC)

Capacidad para comunicar conceptos complejos de Ingeniería, dentro de la profesión y con la sociedad en general. Estas habilidades incluyen: la habilidad de comprender y escribir efectivamente informes, documentación de diseños, realizar presentaciones efectivas, dar y recibir instrucciones claras. Es conveniente incentivar la capacidad de comunicarse en un segundo idioma.
El atributo de habilidades de comunicación será evaluado sumativamente en las secciones de artículo (*paper*) y video de presentación de los resulados de los entregables.

3. Descripción general

El paralelismo a nivel de datos ha tenido históricamente un gran campo de aplicación. Desde los años 70's, el diseño e implementación de arquitecturas vectoriales ha tenido un desarrollo continuo, siendo los procesadores vectoriales la referencia para otros tipos de arquitecturas que utilizan el concepto de *Single-Instruction Multiple-Data* (SIMD), en una gran cantidad de aplicaciones. El desarrollo de arquitecturas heterogéneas, en las que se combinan diferentes tipos de paralelismo, entre ellos el paralelismo a nivel de datos, ha tenido un papel fundamental en los sistemas modernos. Los dispositivos móviles, por ejemplo, hacen uso de arquitecturas SIMD para favorecer el desempeño en ejecución de tareas relacionadas a multimedia, en las que el procesamiento paralelo es fundamental.

Para este proyecto se deberán aplicar los conceptos de arquitectura de computadores, vista como una combinación de elementos de software y hardware, en el diseño e implementación de un modelo en software de procesador vectorial, cuya arquitectura del set de instrucciones (ISA) deberá ser propuesta por cada estudiante, para el desarrollo de una aplicación de encriptación de imágenes, en la que cantidad de datos a procesar crea la necesidad explotar el paralelismo a nivel de datos para lograr un mayor desempeño.
En el proyecto se desarrollará un acercamiento práctico al diseño de un set de instrucciones propio y específico, la realización en software de un modelo funcional de un procesador vectorial que implemente el set propuesto, y programación de sistemas computacionales en general.

4. Especificación

Para este proyecto se deberá diseñar e implementar una arquitectura vectorial para la encriptación de imágenes en escala de grises. La arquitectura incluirá un diseño propio del set de instrucciones (número y tipo de instrucciones, formato de operandos, tamaño, encodificación, modos de direccionamiento, etc), así como una implementación en software de un modelo (simulación) del procesador como tal.
Una vez diseñado e integrado el sistema a nivel de modelo en software, se deberá diseñar una aplicación que aplique cuatro métodos de encriptación de imágenes, que deberán aplicarse a cada uno de los pixeles de la misma. En este punto, la arquitectura vectorial favorecerá el desempeño del sistema, al aplicar la tarea de procesamiento, al menos, a 8 pixeles simultáneamente, por medio de algún mecanismo de implementación paralela de operaciones vectoriales.

A continuación se describe a mayor detalle la especificación del proyecto:

4.1. Arquitectura del set de instrucciones - ISA

Para el desarrollo del proyecto deberá plantearse como punto inicial la arquitectura del set de instrucciones (ISA) que utilizará como base para el diseño del modelo de software del procesador, así como la programación sobre el mismo. El set de instrucciones deberá poseer la documentación adecuada sobre todos los elementos del mismo. Será importante detallar cada una de las instrucciones en cuanto a funcionalidad, sintáxis, modos de direccionamiento, formato, tipo de datos, encodificación, etc. En este punto debe tenerse en cuenta además la cantidad y tipo de registros de propósito general, y la interfaz con memoria (esquema Von Neumann, Hardward). Cada decisión tomada en el set de instrucciones deberá ser justificada con base a la aplicación específica (que se detalla más adelante) y aspectos de eficiencia, tomando en cuenta recursos (costo, área, potencia, etc). Como primer producto del proyecto deberá generarse un documento con la descripción completa y detallada del set, así como un hoja de referencia rápida al set y los aspectos más importantes del mismo. En general, los sets diseñados deberán contar al menos con los siguientes tipos de instrucciones **vectoriales**: operaciones aritméticas, operaciones lógicas, carga, almacenamiento, desplazamientos regulares en ambas direcciones y desplazamientos circulares en ambas direcciones. El tipo de dato del set serán vectores de numeros enteros de 8 bits (el signo, o no, quedará a criterio de cada estudiante con la debida justificación del caso). El tamaño del vector deberá ser de al menos 8 bytes, es decir cada vector deberá tener, al menos, 8 datos de 8 bits cada uno.

Se deberán incluir al menos 12 instrucciones, que deberán incluir operaciones vector-vector y vector-escalar.

4.2. Modelo de la organización

Desde el punto de vista de organización, el procesador que implemente el set deberá utilizar la técnica de segmentación (*pipeline*), para aumentar el desempeño. Para esto, deberá implementarse (de ser el caso) un lógica de detección de riesgos, que puede realizarse por programa o por medio de la organización. Adicionalmente, el procesador debe poseer al menos 4 *lanes* para la ejecución paralela de las operaciones en los vectores. El modelo de sofware debe emular el comportamiento paralelo del hardware, por lo que las abstracciones utilizadas deben soportar concurrencia en cierta medida.

4.3. Aplicación

Desde el punto de vista de programa, se deberá diseñar una aplicación que a partir de una imagen en escala de grises (directamente pre-cargada en memoria, o cargada dinámicamente por algún método) aplique cada uno de los algoritmos de encriptación de imágenes que se describirán adelante, y muestre la imagen original, la imagen encriptada y la imagen desencriptada en una ventana.
Los algoritmos de encriptación serán los siguientes:

- **XOR** con clave privada: Este tipo de encriptación es uno de los más utilizados como base de algoritmos criptográficos más complejos, como AES, por ejemplo. Para este algoritmo al color (grises) de cada pixel (i,j) deberá aplicársele una operación XOR con un dato de 8 bits, denominado clave privada. Para desencriptar una imagen encriptada con este algoritmo, debe aplicarse el mismo proceso.

- **Desplazamiento circular**: Este algoritmo deberá aplicar un desplazamiento circular de una cantidad entre 0-255 a cada pixel, lo que implica que los datos que serán desplazados no se perderán, sino que pasan del bit más significativo al menos significativo, y viceversa. Para desencriptar, se deberá aplicar el desplazamiento circular en la dirección contraria a la encriptación, para la misma cantidad de bits desplazados.

- **Suma simple**: En este algoritmo, al color de cada pixel, dentro de un vector, deberá sumarsele un valor determinado diferente, dentro de otro vector (clave). Así, para un vector de 4 pixeles [30,60,1,1], el vector clave a sumar (para toda la imagen) podrá ser, por ejemplo, [12, 5, 100, 10], y el resultado de color, para este primer vector de pixeles debe ser entonces [42,65,101,11]. Deberá considerarse asuntos de desbordamiento. Para desencriptar, deberá restarse cada vector de pixeles en la imagen con respecto al mismo vector clave, definido previamente.

- **Algoritmo de encriptación propio**: Cada estudiante deberá investigar sobre algoritmos de encriptación de imágenes utilizados en seguridad informática. Luego de dicha investigación, deberá seleccionar e implementar uno para la arquitectura propia.

Adicionalmente se debe diseñar una aplicación/script que convierta de lenguaje ensamblador propio a código máquina (binario). Además, se debe crear un programa que demuestre de manera inequívoca el funcionamiento de todas las instrucciones del set.

4.4. Consideraciones generales

Algunos requerimientos adicionales y consideraciones generales se listan a continuación.

- La elección del lenguaje de programación y demás herramientas de software a utilizar queda a criterio de cada estudiante.
- El modelo a implementar, en software, debe ser una representación funcional correcta de un organización en hardware. En todo momento se podrá visualizar las señales y registros internos del procesador, los contenidos de la memoria, las etapas, datos y señales del pipeline, entre otros componentes de la organización.
- Se recomienda el uso del paradigma de programación orientada a objetos, para dar una mejor representación de concurrencia de los módulos de hardware.
- La herramienta debe mostrar un diagrama de la organización del sistema.
- El modelo diseñado debe permitir las mediciones de tiempo de ejecución, en ciclos de reloj del procesador, así como un estimado de tiempo de ejecución a una frecuencia de 1GHz.
- El modelo debe permitir una ejecución por ciclo, así como una ejecución total (sin pausas). En la ejecución por ciclo se debe poder visualizar todas las señales y contenido de la organización, así como dar seguimiento de los datos en las etapas del pipeline.
- El desarrollo de software deberá realizarse utilizando un repositorio en línea como sistema de control de versiones.
- Para la codificación del software deberá establecerse explícitamente (en la sección de diseño) y seguirse adecuadamente algún código, norma o estándar establecido.

4.5. Notas adicionales

- El desarrollo de este proyecto se dará individual.
- Todo diseño deberá tener al menos 2 propuestas detalladas adecuadamente y comparadas según criterios.

5. Entregables

Como entregables en este proyecto se evaluará lo siguiente:

- Presentación funcional completa (65 %): Para la presentación funcional, cada estudiante deberá grabar un video realizando las diferentes pruebas sobre la herramienta (grabación de la pantalla) y explicando el funcionamiento interno (etapas/señales/etc) y limitantes. El profesor evaluará las pruebas según rúbrica correspondiente. En caso de que haya duda de cualquiera de las partes, el estudiante o el profesor podrá solicitar una cita para la

defensa del proyecto de forma presencial. Se deberá subir un archivo con el video o un link al video correspondiente junto con un enlace al repositorio del código fuente, en la sección de evalaciones del TecDigital.

- Artículo tipo *paper* (10 %): El paper a realizar deberá tener una extensión no mayor a 4 páginas, deberá ser realizado con LaTeX, siguiendo un formato establecido (IEEE Transactions o ACM, por ejemplo). En general el *paper* deberá contar con las siguientes secciones:

  - Resumen e introducción - 2 %
  - Sistema desarrollado - 2 %
  - Resultados - 2 %
  - Conclusiones - 2 %
  - Referencias - 2 %

- Presentación del *paper* y resultados (5 %): Cada persona deberá grabar un video de no más de 5 minutos, presentando la idea de su solución (puede utilizar diapositivas o algún otro medio de referencia). Debe considerar que el público meta de su presentación no tiene necesariamente el *background* técnico, por lo que deberá exponer de forma clara lo que se ha realizado, así como los resultados más importantes de su diseño. Puede utilizar el canal de Youtube "Two Minute Papers"`https://www.youtube.com/user/keeroyz`, como referencia.

- Documentación de diseño (20 %): La documentación del diseño deberá contener las siguientes secciones:

  - Listado de requerimientos del sistema: Cada estudiante deberá determinar los requerimientos de ingeniería del problema planteado, considerando partes involucradas, estado del arte, estándares, normas, entre otros.
  - Elaboración de opciones de solución al problema: Para el problema planteado deberán documentarse al menos dos opciones de solución. Cada solución deberá ser acompañada de algún tipo de diagrama.
  - Comparación de opciones de solución: Se deberán comparar explícitamente las opciones de solución, de acuerdo con los requerimientos y otros aspectos aplicables de salud, seguridad, ambientales, económicos, culturales, sociales y de estándares.
  - Selección de la propuesta final: Se deberá seleccionar una propuesta final de las opciones de solución, de acuerdo con los criterios de comparación.
  - Implementación del diseño: Se deberá documentar completamente el diseño final seleccionado. Para el caso de este proyecto esto incluye: descripción de arquitectura del set de instrucciones (ISA), diagrama de bloques del modelo del procesador, diagrama de bloques del computador (procesador + interfaz con aplicación), diagramas propios de diseño de sofware aplicables (de flujo, clases, composición, UML, patrones de diseño, etc) y descripción de algoritmo propuesto.